Conference on

Machine Processing of

Remotely Sensed Data

October 16 - 18, 1973

The Laboratory for Applications of
Remote Sensing

Purdue University
West Lafayette
Indiana

THE USE OF THE MODIFIED CHOLESKY DECOMPOSITION IN DIVERGENCE AND CLASSIFICATION CALCULATIONS†

D. L. Van Rooy, M. S. Lynn and
C. H. Snyder

Rice University
Houston, Texas

ABSTRACT

This report analyzes the use of the modified Cholesky decomposition technique as applied to the feature selection and classification algorithms used in the analysis of remote sensing data (e. g. as in LARSYS). This technique is approximately 30% faster in classification and a factor of 2-3 faster in divergence, as compared with LARSYS. Also numerical stability and accuracy are slightly improved. Other methods necessary to deal with numerical stability problems are briefly discussed.

1. INTRODUCTION

This report analyzes the use of the Cholesky decomposition[1]* technique in the analysis of remotely sensed data, specifically in divergence calculations and in the evaluation of the maximum likelihood function; the latter occur in, respectively, the feature selection and classification techniques used, for example, in the LARSYS[2] system developed by the Laboratory for the Applications of Remote Sensing of Purdue University.

Although LARSYS was primarily developed for research purposes, increasing use of the system and of derivative systems such as ERIPS[3] for production processing emphasizes the need for efficient, accurate and stable algorithms as the basis for design objectives of computational analysis. The organization of computation in certain segments of LARSYS and the use of subroutines such as MINV from the IBM Scientific Subroutine Package (SSP)[4] do not lend credence that such design objectives have been met. The purpose of this report is to describe how one possible re-organization of the computation and the use of preferred techniques can improve the efficiency and accuracy of the system.

The focus of this report is on improved efficiency in terms of computation time. Thus the arithmetic precision used is identical with that used in LARSYS, so that a meaningful comparison of efficiency can be obtained. It will be shown that the algorithms proposed yield improvements in computational speed with no loss in accuracy or stability (in fact, slight improvements can be obtained in the latter).

† Research supported under NASA contract NAS-9-12776

* Numbers in superscripts refer to references

Improvements in accuracy and stability can be achieved by further refinements in the techniques used. This will be the subject of a later report; however, in Section 6, we discuss where such improvements can be expected by the use of higher precision and/or the use of such techniques as iterative refinement, scaling and equilibration.

It is, in our view, extremely important that the best numerical techniques be used in production calculations. The argument that sub-optimal techniques have sufficed in the past is not valid if one considers that unexpected failures in the future may be extremely costly to rectify; since the use of the validated techniques discussed in this report are both more reliable and efficient, it would seem wiser to proceed into future production calculations with the assurance that the systems and methods used rest on a more secure algorithmic foundation.

2. CHOLESKY DECOMPOSITION

Let K be real, nxn, symmetric positive-definite matrix. In the applications under consideration, K would be a covariance matrix. Then there is a unique, nxn, real, lower-triangular matrix, L, such that (Cholesky decomposition)

$$K = LL^* \tag{2.1}$$

where L^* denotes the (conjugate) transpose of L. There is also a unique, real, lower-triangular matrix, $\tilde{L}$, and a real, positive diagonal matrix, $\tilde{D}$, such that (modified Cholesky decomposition)

$$K = \tilde{L}\tilde{D}L^* \tag{2.2}$$

where $\tilde{L}$ has diagonal elements equal to unity. From (2.1) and (2.2) it can be seen that

$$L = \tilde{L}\tilde{D}^{\frac{1}{2}} \tag{2.3}$$

where $\tilde{D}^{\frac{1}{2}}$ is the diagonal matrix whose entries are the square roots of the corresponding elements of $\tilde{D}$.

Either the Cholesky or modified Cholesky decompositions can be readily obtained from the following recurrence relationships[1], [5] (we use the notation $K = (k_{ij})$, $L = (\ell_{ij})$, $\tilde{L} = (\tilde{\ell}_{ij})$, $\tilde{D} = \text{diag}\ \{\tilde{d}_i\}$):

Cholesky

$$\left.\begin{aligned} \ell_{11} &= k_{11}^{\frac{1}{2}} \\ \ell_{jj} &= \left(k_{jj} - \sum_{s=1}^{j-1} \ell_{js}^2\right)^{\frac{1}{2}} \\ \ell_{ij} &= \left(k_{ij} - \sum_{s=1}^{j-1} \ell_{is}\,\ell_{js}\right)\Big/\ell_{jj} \end{aligned}\right\} \quad \begin{aligned} &j = 1, \ldots, n \\ &i = j+1, j+2, \ldots, n \end{aligned} \tag{2.4}$$

and, of course, $\ell_{ij} = 0$ for $j > i$.

Modified Cholesky

$$\left.\begin{aligned} \tilde{d}_1 &= k_{11} \\ \tilde{d}_j &= k_{jj} - \sum_{s=1}^{j-1} \tilde{d}_s \, \tilde{\ell}_{js}^2 \\ \tilde{\ell}_{ij} &= \Big(k_{ij} - \sum_{s=1}^{j-1} \tilde{d}_s \, \tilde{\ell}_{is} \, \tilde{\ell}_{js}\Big)/\tilde{d}_j \\ & \qquad i = j+1, \ldots, n \end{aligned}\right\} \quad j = 1, \ldots, n \tag{2.5}$$

where $\tilde{\ell}_{ii} = 1$ $(i=1, \ldots, n)$ and $\tilde{\ell}_{ij} = 0$ for $j > i$.

For the applications under consideration, the modified Cholesky decomposition is more useful since it avoids the computation of square roots inherent in (2.4). It can easily be shown that, under the assumption that K is positive-definite, $\tilde{d}_j > 0$ $(j=1, \ldots, n)$.

Once either decomposition is obtained, solutions of equations of the form

$$Kx = b \tag{2.6}$$

may readily be obtained from the back and forward substitutions (we henceforth only consider the modified Cholesky decomposition) :

$$y_1 = \tilde{L}^{-1} b \tag{2.7}$$

$$y_2 = \tilde{D}^{-1} y_1 \tag{2.8}$$

$$x = \tilde{L}^{*-1} y_2 \tag{2.9}$$

since

$$\begin{aligned} Kx &= \tilde{L}\tilde{D}\tilde{L}^* x \\ &= \tilde{L}\tilde{D}y_2 \\ &= \tilde{L}y_1 \\ &= b \end{aligned}$$

as desired. (2.7) - (2.9) may alternatively be written (using $\leftarrow$ to denote replacement as opposed to equality) to economize on storage :

$$x_1 \leftarrow b_1 / \tilde{d}_1$$

$$x_i \leftarrow \left(b_i - \sum_{j=1}^{i-1} \tilde{\ell}_{ij}\, \tilde{d}_j\, x_j \right) / \tilde{d}_i$$

$$i = 2, \ldots, n \qquad (2.10)$$

$$x_i \leftarrow \left(x_i - \sum_{j=i+1}^{n} \tilde{\ell}_{ji}\, x_j \right)$$

$$i = n-1, n-2, \ldots, 1$$

Note that in order to solve such systems there is no requirement to calculate K^{-1}, only $\tilde{L}$ and $\tilde{D}$ which requires approximately 1/3 the amount of computation.

This saving in itself is significant if one considers that the amount of time devoted to computing matrix inverses in connection with feature extraction in LARSYS varies roughly as mn^3, where m is the number of classes and n is the number of features under consideration - the corresponding amount of time devoted to the actual divergence calculation varies as $1/2m^2n^2$, which is of the same order of magnitude for most problems considered. Thus reducing the first factor by a third can significantly effect the overall computation time of itself.

In the applications under consideration, we thus have m covariance matrices K_s ($s=1,\ldots,m$) corresponding to the number of classes. The dimensionality, n, of each K_s corresponds to the number of channels. With obvious notation, we write

$$K_s = \tilde{L}_s \tilde{D}_s \tilde{L}_s^* \qquad s=1, \ldots, m$$

where

$$K_s = (k_{ij}^{(s)}), \quad \tilde{L}_s = (\tilde{\ell}_{ij}^{(s)}), \quad \tilde{D}_s = \text{diag}\left\{\tilde{d}_i^{(s)}\right\}$$

and $\left\{\tilde{\ell}_{ij}^{(s)}\right\}$, $\left\{\tilde{d}_i^{(s)}\right\}$ are calculated as in (2.5)

3. FEATURE SELECTION

Feature selection, as implemented in LARSYS, depends upon calculating a measure of inter-class divergence for multiple classes, requiring calculations of the form

$$D = D_1 + D_2 \qquad (3.1)$$

where

$$D_1 = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \text{tr}\left[(K_i - K_j)(K_j^{-1} - K_i^{-1})\right] \qquad (3.2)$$

$$D_2 = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (u^i - u^j) * (K_i^{-1} + K_j^{-1})(u^i - u^j) \qquad (3.3)$$

where tr A denotes the trace of A (sum of its diagonal elements) and u^s (s=1 , ... , m) is the mean vector for the s^{th} class. We first simplify (3. 2) and (3. 3)

We note that we can write

$$D_1 = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (\mathrm{tr} K_i K_j^{-1}) + \sum_{j=1}^{m-1} \sum_{i=j+1}^{m} (\mathrm{tr} K_i K_j^{-1}) - nm(m-1)$$

$$= \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (\mathrm{tr} K_i K_j^{-1}) + \sum_{i=2}^{m} \sum_{j=1}^{i-1} (\mathrm{tr} K_i K_j^{-1}) - nm(m-1)$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{m} \mathrm{tr}(K_i K_j^{-1}) - nm^2$$

$$= \sum_{j=1}^{m} \sum_{i=1}^{m} \mathrm{tr}(K_j^{-1} K_i) - nm^2$$

(since tr(AB) = tr(BA))

$$= \sum_{j=1}^{m} \mathrm{tr}(K_j^{-1} K) - nm^2 \qquad (3.4)$$

where

$$K = \sum_{i=1}^{m} K_i$$

$$= \tilde{L}\tilde{D}\tilde{L}^* \qquad \text{(say)}$$

Now

$$\mathrm{tr} K_j^{-1} K = \mathrm{tr}(\tilde{L}_j^{*-1} \tilde{D}_j^{-1} \tilde{L}_j^{-1} \tilde{L}\tilde{D}\tilde{L}^*)$$

$$= \mathrm{tr}(\tilde{D}_j^{-1} T_j \tilde{D} T_j^*)$$

where

$$T_j = \tilde{L}_j^{-1} \tilde{L} = (t_{ij}) \qquad \text{(say)} \qquad (3.5)$$

Thus

$$\mathrm{tr}(K_j^{-1} K) = \sum_{p=1}^{n} \sum_{q=1}^{p} (t_{pq}^{(j)2} / \tilde{d}_p^{(j)}) \tilde{d}_q \qquad (3.6)$$

$$j=1 , \ldots , m$$

Hence D_1 may economically be calculated from (3.4), (3.5) and (3.6). It should be noted that the calculation of the $\{T_j\}$ in (3.5) each require n calculations of the form (2.10); however, since $\tilde{T}_j, \tilde{L}_j$ and $\tilde{L}$ are all lower triangular, it is important to remark that much of the computation may be reduced by observing that, in calculating the q^{th} column of T_j, the index n in (2.10) is actually replaced by $n-q+1$ $(q=1,\ldots,n)$.

The calculation of D_2 may be similarily simplified. For, from (3.3), we may write

$$D_2 = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (u^i - u^j)^* (K_i^{-1} + K_j^{-1}) (u^i - u^j)$$

$$= \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (u^i - u^j)^* K_i^{-1} (u^i - u^j)$$

$$+ \sum_{j=1}^{m-1} \sum_{i=j+1}^{m} (u^i - u^j)^* K_j^{-1} (u^i - u^j) \qquad (3.7)$$

(interchanging i and j in the second sum). Interchanging the order of subscripts gives

$$D_2 = \sum_{i=1}^{m} \sum_{j=1}^{m} (u^i - u^j)^* K_i^{-1} (u^i - u^j)$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{m} \eta^{ij*} \tilde{D}_i^{-1} \eta^{ij} \qquad (3.8)$$

where

$$\eta^{ij} = \tilde{L}_i^{-1} (u^i - u^j)$$

$$= \delta^{ii} - \delta^{ij} \qquad i, j = 1, \ldots, m$$

where the computation of

$$\delta^{ij} = \tilde{L}_i^{-1} u^j \qquad (3.9)$$

involves a forward substitution, that is δ^{ij} is obtained from

$$\delta_1^{(ij)} = u_1^{(j)}$$

$$\delta_p^{(ij)} = u_p^{(j)} - \sum_{q=1}^{p-1} \tilde{\ell}_{pq}^{(i)} \delta_q^{(ij)}, \qquad p=2,\ldots,n \qquad (3.10)$$

We thus have, from (3.8) that

$$D_2 = \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{p=1}^{n} (\eta_p^{(ij)})^2 / \tilde{d}_p^{(i)} \tag{3.11}$$

where

$$\eta_p^{(ij)} = \delta_p^{(ii)} - \delta_p^{(ij)}$$

and the $\{ \delta_p^{(ij)} \}$ are calculated from (3.10).

The use of the above formulae should probably not be compared with the approach used in LARSYS itself, but with the improvements proposed by G. Austin[8] which take full advantage of the symmetry of the $\{K_i\}$ and of the symmetric structure of the summands in (3.7). It can be shown that the amount of work involved in calculating D_2 in (3.11) is comparable with that involved with the corresponding terms in Ref. (8). However, the amount of work involved in evaluating (3.4) is actually considerably less than the method proposed in Ref. (8) on account of the asymptotic linear dependence on m, as opposed to the quadratic dependence of Ref. (8). It should nevertheless be pointed out that, from (3.4)

$$D_1 = \mathrm{tr}(\hat{K}K) - nm^2 \tag{3.12}$$

where

$$\hat{K} = \sum_{j=1}^{m} K_j^{-1} \tag{3.13}$$

Thus, if the K_j^{-1} have been precomputed, the amount of work involved in evaluating D_1 may become negligible compared with the evaluation of D_2 by using (3.12) and the fact that, for symmetric matrices A, B:

$$\mathrm{tr}\,(AB) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{ji}$$

$$= 2 \sum_{i=1}^{n} \sum_{j=1}^{i-1} a_{ij} b_{ji} + \sum_{i=1}^{n} a_{ii} b_{ii}$$

However, this approach does not obviate the overall savings in feature selection of using the Cholesky decomposition instead of computing matrix inverses.

4. CLASSIFICATION

Classification involves the calculation of the maximum likelihood functions

$$f_j(x) = \sigma_n \alpha_j \exp\left[-\tfrac{1}{2} (x - u^j)^* K_j^{-1} (x - u^j) \right]$$

$$j = 1, \ldots, m \tag{4.1}$$

where x is the observation vector, $\sigma_n = 1/(2\pi)^{n/2}$, and

$$\alpha_j = (\det K_j^{-1})^{\frac{1}{2}}$$

$$= \left(\prod_{p=1}^{n} \tilde{d}_p^{(j)}\right)^{-\frac{1}{2}} \qquad (4.2)$$

Actually, since $\exp(x)$ is a monotonic increasing function, only $\log f_j(x)$ needs to be computed in determing the maximum of $f_j(x)$ over all m classes.

However, (4.1) is again simplified by noting that

$$(x - u^j)^* K_j^{-1} (x - u^j) = y_j^* \tilde{D}_j^{-1} y_j$$

where

$$y_j = \tilde{L}_j^{-1} (x - u^j)$$

is calculated in a manner analogous to (3.9).

5. RESULTS

The above techniques have been tested by appropriately modifying the OS version of LARSYS[6] supplied by NASA-JSC. In actuality the modification to the divergence calculations in feature selection use the Cholesky decomposition as opposed to the modified Cholesky decomposition as discussed in Section 3 - further savings of time, obtained by not having to calculate square roots, could be realized by using the modified Cholesky decomposition.

The modifications were written in single-precision FORTRAN and compared with the original single-precision versions in LARSYS. In the case of classification, the results were also compared with a single-precision version of the corresponding calculations in LARSYS written in assembly code.

The precision of these timing results is very open to question due to the difficulty of obtaining accurate and reproducible timing information under the OS Operating System of the IBM 370/155. Timings are heavily dependent on general system activity; furthermore the considerable subroutine overhead inherent to the computation tends to mask much of the potential arithmetic economies of efficiency.

The results are summarized in Figures 1, 2 and 3 on test data supplied by Purdue University with LARSYS. Figure 1, depicts the ratio of the time taken by the original LARSYS version (DIVERG) to that taken by the proposed algorithm (CHOLESKY) in a divergence calculation for feature selection using six channels; this ratio is plotted for a varying number of target classes. It can be seen that CHOLESKY is approximately twice as fast as DIVERG.

Theoretical analysis shows that this ratio should be greater than three for all values of m, and asymptotically should approach four for large values of m. This discrepancy underscores the high degree of imprecision associated with the timing results.

In Figure 2, the same ratio is plotted for a fixed number of classes (11) and where a varying number of features is selected from twelve channels. Except for a very small number

of features, where the order of the K_j is so small that the time of calculation is dominated by computational overhead, it can again be seen that CHOLESKY is between two and three times faster than DIVERG. Again, theoretically, this ratio should be between three and four for all values of the number of features.

In Figure 3, the time taken for classification using the three methods is compared for a number of points varying from 50,000 to 100,000. The Cholesky method is significantly faster (about 30%). It should be pointed out that, as has been noted elsewhere[7], equivalent savings can be obtained by using a variant of the LARSYS calculations which does not employ the modified Cholesky decomposition; however, this variant does not have the accuracy potential of the Cholesky approach[1].

6. IMPROVEMENTS IN ACCURACY

The modifications described were executed in single-precision so as to provide a basis for comparison with the LARSYS calculations. Without further refinement, it should not be surprising that the accuracy will be correspondingly limited, since[1] accuracy in such computations is essentially a function of three principal components:

- the method employed
- the arithmetic significance
- the conditioning of the various matrices

For ill-conditioned systems (in the applications under consideration, these may arise, for example, from working with highly-correlated channels), more precise methods have to be employed and/or the arithmetic significance increased. Directions which need to be examined with higher accuracy objectives in mind include, not only that of using higher significance arithmetic in sensitive portions of the computation, but also those of employing iterative refinement, scaling or equilibration. These will, however, be studied in a later report.

REFERENCES

(1) G. Forsythe & C.B. Moler, Computer Solution of Linear Algebraic Systems, Prentice Hall, Englewood Cliffs, N. J. 1967.

(2) Supplied by LARS, Purdue Univ., West Lafayette, Indiana.

(3) Earth Resources Interactive Processing System (ERIPS) prepared by IBM Federal Systems Division for NASA - JSC.

(4) System 360 Scientific Subroutine Package, Version III, IBM Application Program.

(5) A. Ralston and H.S. Wilf, Mathematical Methods for Digital Computers, Vol. II, John Wiley and Sons, New York, 1967.

(6) The OS version of LARSYS was written by IBM - Federal Systems Division for NASA - JSC.

(7) G. Austin, "Analysis of LARS Subroutine CLASS and Recommended Coding Improvements to Reduce Its Execution Time," NASA - JSC MPAD Memorandum, May 16, 1972.

(8) G. Austin, "Modifications to ERIPS Requirements to Reduce Computation Time and Storage Requirements," MSC Internal Note No. 72-FM-210, Jan. 16, 1973 NASA.

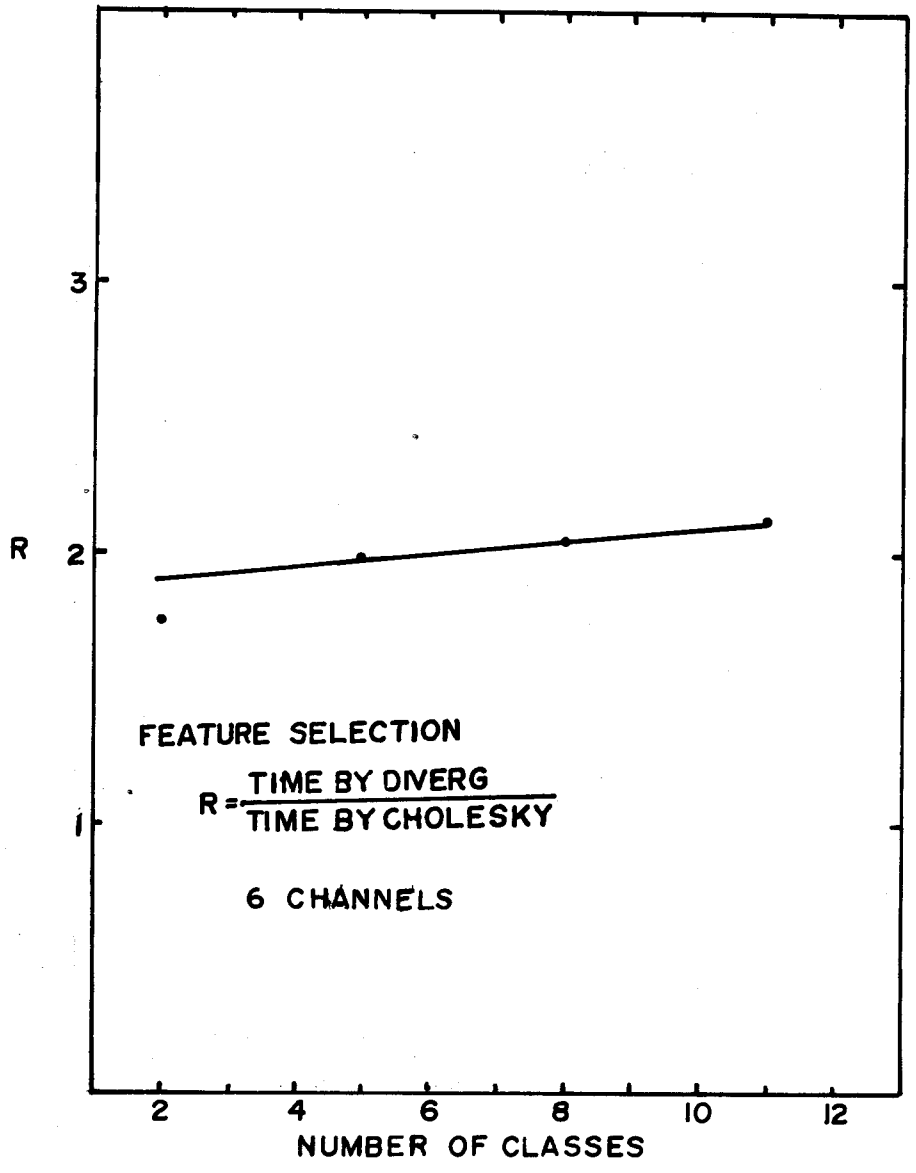

Figure 1. Timing comparison as a function of number of classes used for the divergence calculation.

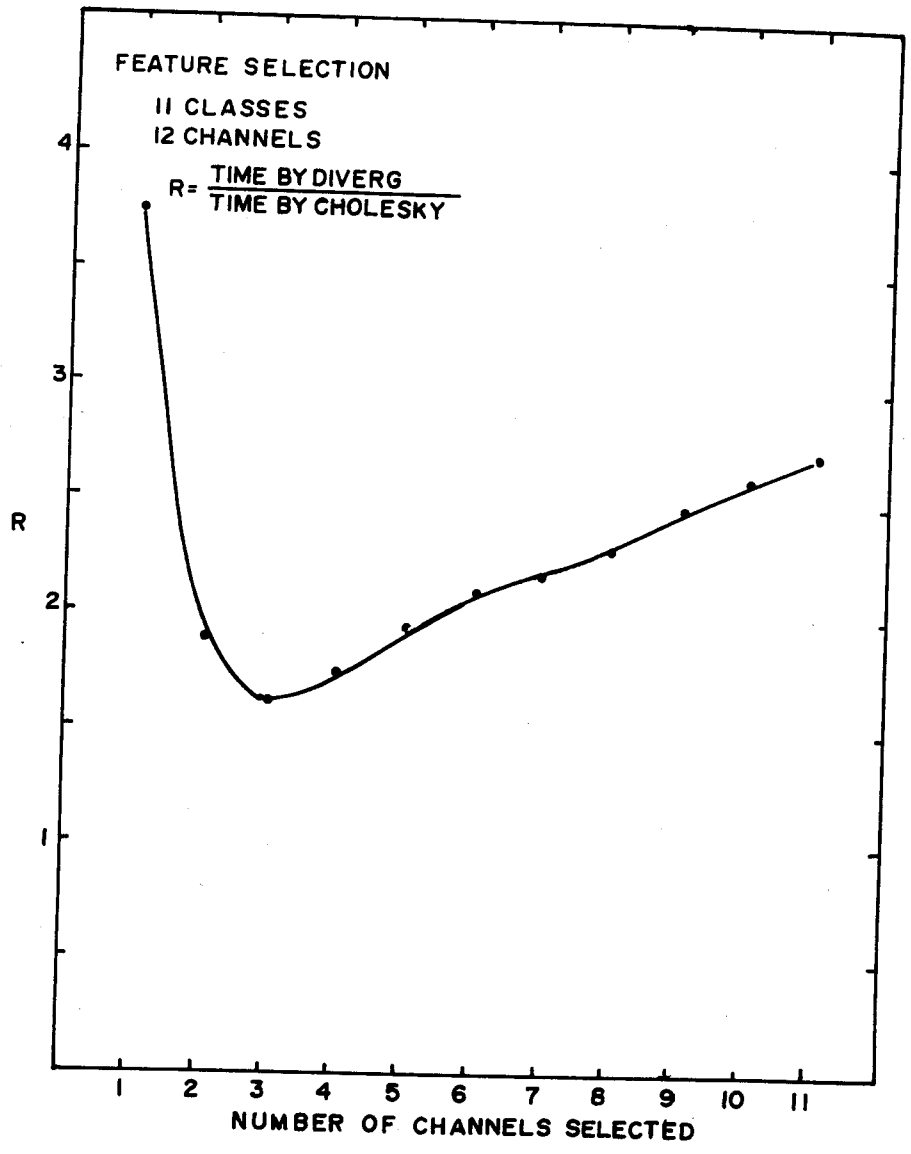

Figure 2. Timing comparison as a function of number of channels selected for the divergence calculation.

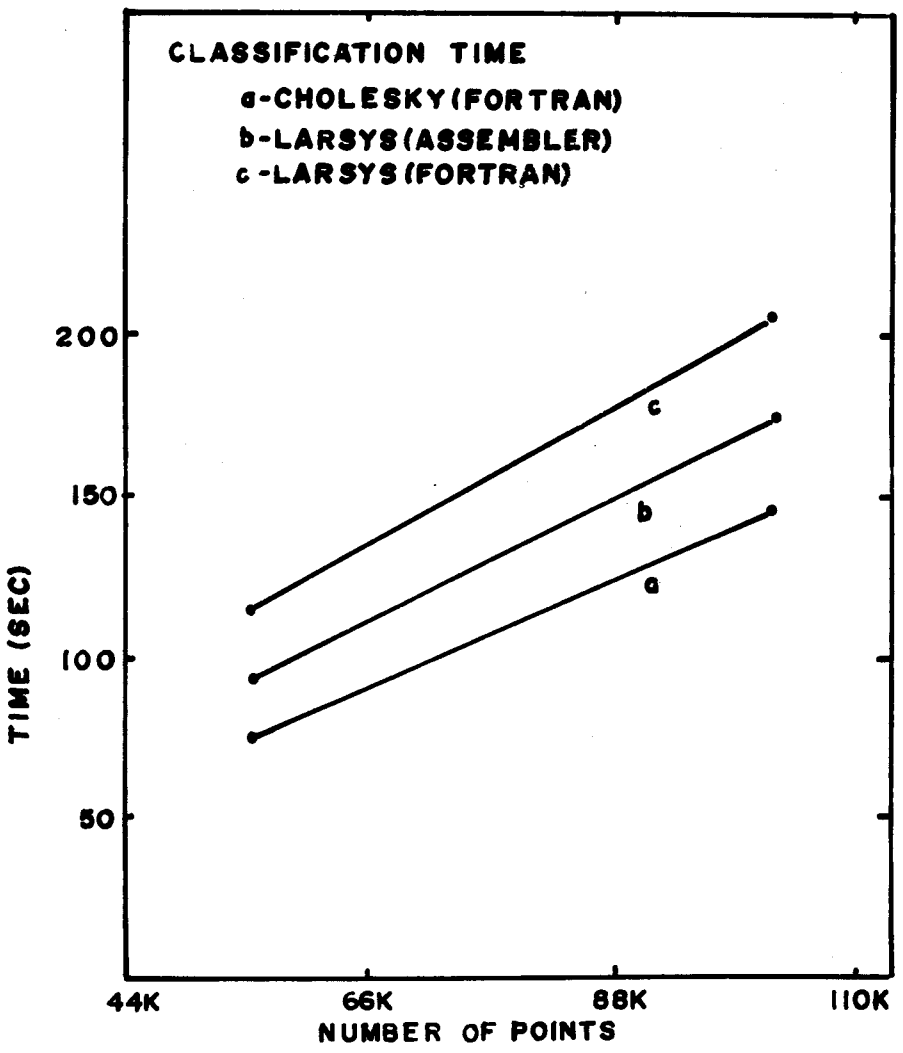

Figure 3. Timing comparison for classification programs.